UNITED STATES DISTRICT COURT

EASTERN DISTRICT OF TEXAS

TIVO INC., a Delaware corporation, Plaintiff, vs. 1. ECHOSTAR COMMUNICATIONS CORPORATION, a Nevada corporation, and 2. ECHOSTAR DBS CORPORATION, a Colorado corporation, Defendant.	)	Case No. 2-04CV-01 DF **DEMAND FOR JURY TRIAL**

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff TiVo Inc. ("TiVo") alleges as follows:

PARTIES

1. TiVo is a Delaware corporation with a principal place of business at 2160 Gold Street, Alviso, California 95002-2160.

2. EchoStar Communications Corporation ("EchoStar") is a Nevada corporation with a principal place of business at 5701 South Sante Fe Drive, Littleton, Colorado, 80120.

3. EchoStar DBS Corporation ("EchoStar DBS") is a Colorado corporation, and a wholly-owned subsidiary of EchoStar. On information and belief, DBS's principal place of business is the same as EchoStar's. Collectively, EchoStar and EchoStar DBS are referred to herein as "Defendants."

JURISDICTION

TIVO Inc v. Echostar Comm, et al Doc. 1

4. This is an action for patent infringement arising under the patent laws of the United States of America, 35 U.S.C. section 1, et seq., including 35 U.S.C. section 271. This Court has subject matter jurisdiction pursuant to 28 U.S.C. sections 1331 and 1338(a) in that this is a civil action arising out of the patent laws of the United States of America.

Dockets.Justia.com

VENUE

5. Venue in the Eastern District of Texas is proper pursuant to 28 U.S.C. sections 1391(b) and (c) because, on information and belief, a substantial part of the events or omissions giving rise to TiVo's claims occurred in this district, and Defendants may be found and/or reside in this district.

COUNT ONE
(Patent Infringement)

6. TiVo incorporates by reference paragraphs 1 through 5 of this Complaint and realleges them as though fully set forth herein.

7. On May 15, 2001, the United States Patent and Trademark Office issued U.S. Patent No. 6,233,389, entitled "Multimedia Time Warping System" (the "'389 Patent"), a copy of which is attached as Exhibit A.

8. TiVo is the owner by assignment of all rights, title and interest in and to the '389 Patent.

9. On information and belief, Defendants are making, using, offering to sell and/or selling digital video recording devices, digital video recording device software and/or personal television services in the United States of America.

10. On information and belief, Defendants have been and are now willfully and deliberately infringing, contributorily infringing and/or actively inducing infringement of the '389 Patent by making, using, offering to sell and/or selling in the United States of America digital video recording devices, digital video recording device software and/or personal television services falling within the scope of one or more claims of the '389 Patent.

11. As a consequence of Defendants' infringement, TiVo is entitled to recover damages adequate to compensate it for the infringement complained of herein, but in no event less than a reasonable royalty.

12. The infringement by Defendants has injured and will continue to injure TiVo, unless and until such infringement is enjoined by this Court.

PRAYER FOR RELIEF

WHEREFORE, TiVo requests that the Court:

A. Adjudge that the '389 Patent is valid and enforceable;

B. Adjudge that Defendants are infringing and have infringed, and have contributed to and induced infringement of, the '389 Patent, and that such infringement has been willful and deliberate;

C. Enjoin Defendants and their affiliates, subsidiaries, officers, directors, employees, agents, representatives, licensees, successors, assigns and all those acting for them and on their behalf, or acting in concert with them, from further infringement of the '389 Patent;

D. Award compensatory damages to TiVo, together with interest;

E. Award treble damages to TiVo for Defendants' willful infringement of the '389 Patent;

F. Award TiVo its costs and attorneys' fees pursuant to 35 U.S.C. section 285;

G. Award TiVo such other and further relief as it deems just and proper.

DEMAND FOR JURY TRIAL

Plaintiff TiVo hereby demands a jury trial on all issues so triable.

Dated: January 5, 2004

McKOOL SMITH, P.C.

By: [signature]

Sam Baxter, Attorney-in-Charge
State Bar No. 01938000
sbaxter@mckoolsmith.com

P.O. Box O
505 E. Travis, Suite 105
Marshall, Texas 75670
Telephone: (903) 927-2111
Telecopier: (903) 927-2622

Attorneys for Plaintiff TiVo INC.

Co-counsel:

IRELL & MANELLA LLP
Morgan Chu (70446)
Perry M. Goldberg (168976)
1800 Avenue of the Stars, Suite 900
Los Angeles, California 90067-4276
Telephone: (310) 277-1010
Facsimile: (310) 203-7199

US006233389B1

(12) United States Patent

Barton et al.

(10) Patent No.: US 6,233,389 B1
(45) Date of Patent: May 15, 2001

(54) **MULTIMEDIA TIME WARPING SYSTEM**

(75) Inventors: **James M. Barton**, Los Gatos; **Roderick James McInnis**, Milpitas; **Alan S. Moskowitz**, San Francisco; **Andrew Martin Goodman**, Menlo Park; **Ching Tong Chow**, Fremont; **Jean Swey Kao**, Cupertino, all of CA (US)

(73) Assignee: **TiVo, Inc.**, Alviso, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/126,071**

(22) Filed: **Jul. 30, 1998**

(51) **Int. Cl.**[7] **H04N 5/92**

(52) **U.S. Cl.** **386/46**; 386/68

(58) **Field of Search** 386/1, 33, 45, 386/46, 111–112, 125–126, 68; 369/60; 366/7, 33; 348/7, 10, 571, 714, 722, 725; H04N 5/76, 5/92, 9/79, 5/14

(56) **References Cited**

U.S. PATENT DOCUMENTS

Re. 33,535	2/1991	Cooper	358/149
4,313,135	1/1982	Cooper	358/149
4,665,431	5/1987	Cooper	358/145
5,202,761	4/1993	Cooper	358/149
5,371,551	12/1994	Logan et al.	348/571
5,438,423	8/1995	Lynch et al.	358/335
5,550,594	8/1996	Cooper et al.	348/513
5,572,261	11/1996	Cooper	348/512

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

0726574	8/1996	(EP)	G11B/27/034

OTHER PUBLICATIONS

ASTARTE DVDirector™ Beta Testing Program.

Primary Examiner—Thai Tran
(74) *Attorney, Agent, or Firm*—Michael A. Glenn; Kirk Wong

(57) **ABSTRACT**

A multimedia time warping system. The invention allows the user to store selected television broadcast programs while the user is simultaneously watching or reviewing another program. A preferred embodiment of the invention accepts television (TV) input streams in a multitude of forms, for example, National Television Standards Committee (NTSC) or PAL broadcast, and digital forms such as Digital Satellite System (DSS), Digital Broadcast Services (DBS), or Advanced Television Standards Committee (ATSC). The TV streams are converted to an Moving Pictures Experts Group (MPEG) formatted stream for internal transfer and manipulation and are parsed and separated it into video and audio components. The components are stored in temporary buffers. Events are recorded that indicate the type of component that has been found, where it is located, and when it occurred. The program logic is notified that an event has occurred and the data is extracted from the buffers. The parser and event buffer decouple the CPU from having to parse the MPEG stream and from the real time nature of the data streams which allows for slower CPU and bus speeds and translate to lower system costs. The video and audio components are stored on a storage device and when the program is requested for display, the video and audio components are extracted from the storage device and reassembled into an MPEG stream which is sent to a decoder. The decoder converts the MPEG stream into TV output signals and delivers the TV output signals to a TV receiver. User control commands are accepted and sent through the system. These commands affect the flow of said MPEG stream and allow the user to view stored programs with at least the following functions: reverse, fast forward, play, pause, index, fast/slow reverse play, and fast/slow play.

61 Claims, 12 Drawing Sheets

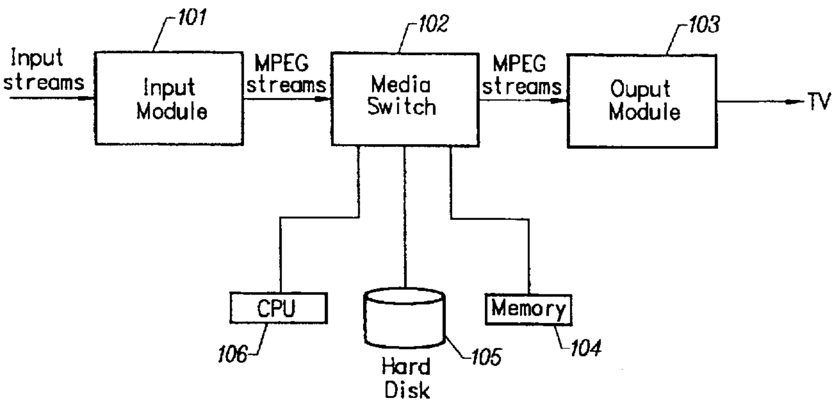


EXHIBIT
A

U.S. PATENT DOCUMENTS

5,675,388		10/1997	Cooper	348/461
5,696,868		12/1997	Kim et al.	386/46
5,706,388		1/1998	Isaka	386/125
5,787,225	*	7/1998	Honjo	386/111
5,920,842		7/1999	Cooper et al.	704/503
5,937,138	*	8/1999	Fukuda et al.	386/112

* cited by examiner

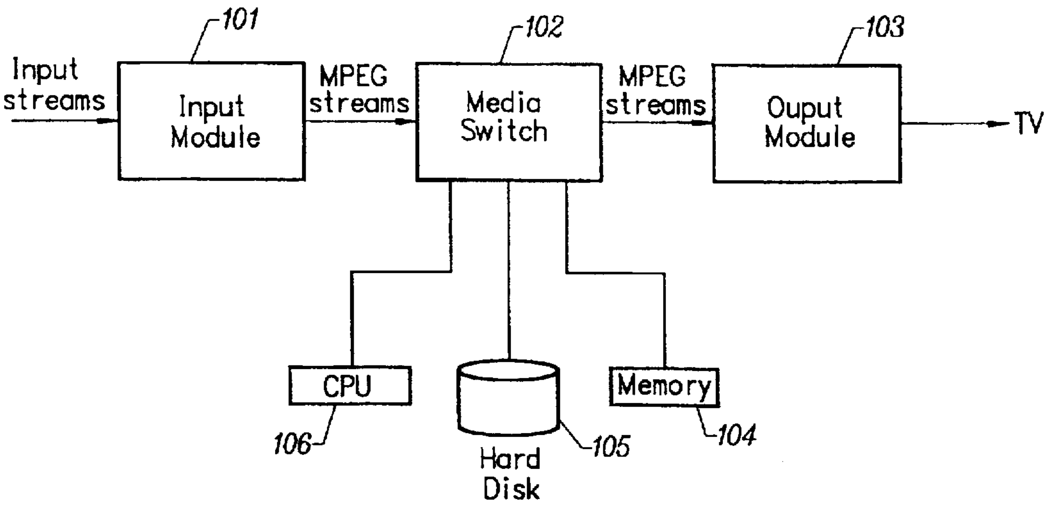

101
102
103
Input streams
Input Module
MPEG streams
Media Switch
MPEG streams
Ouput Module
TV
CPU
Memory
Hard Disk
106
105
104

FIG. 1

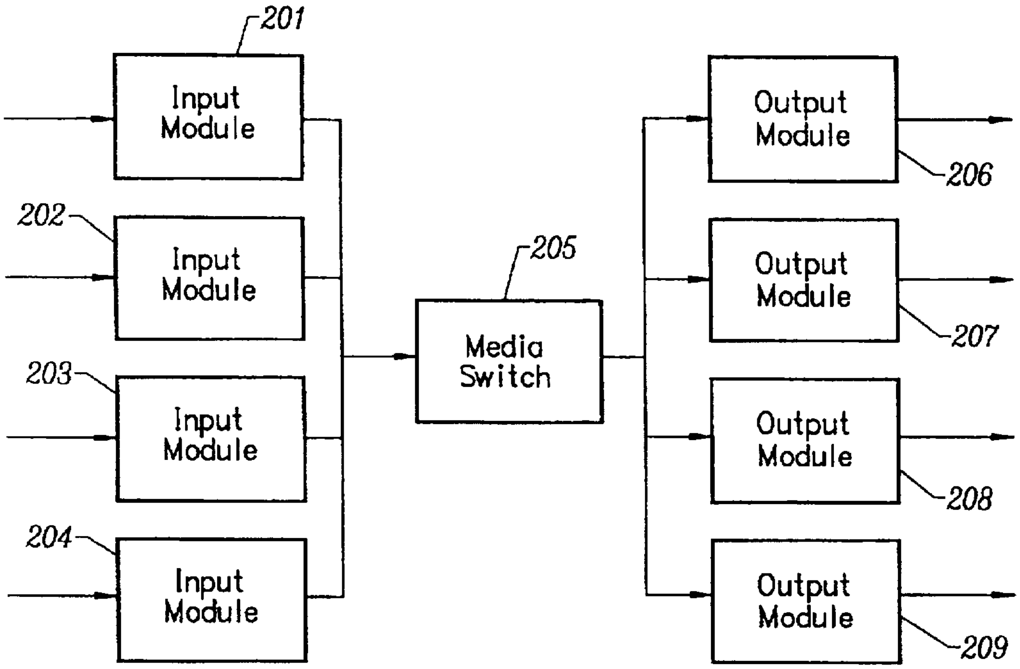

201
Input Module
202
Input Module
203
Input Module
204
Input Module
205
Media Switch
Output Module
206
Output Module
207
Output Module
208
Output Module
209

FIG. 2

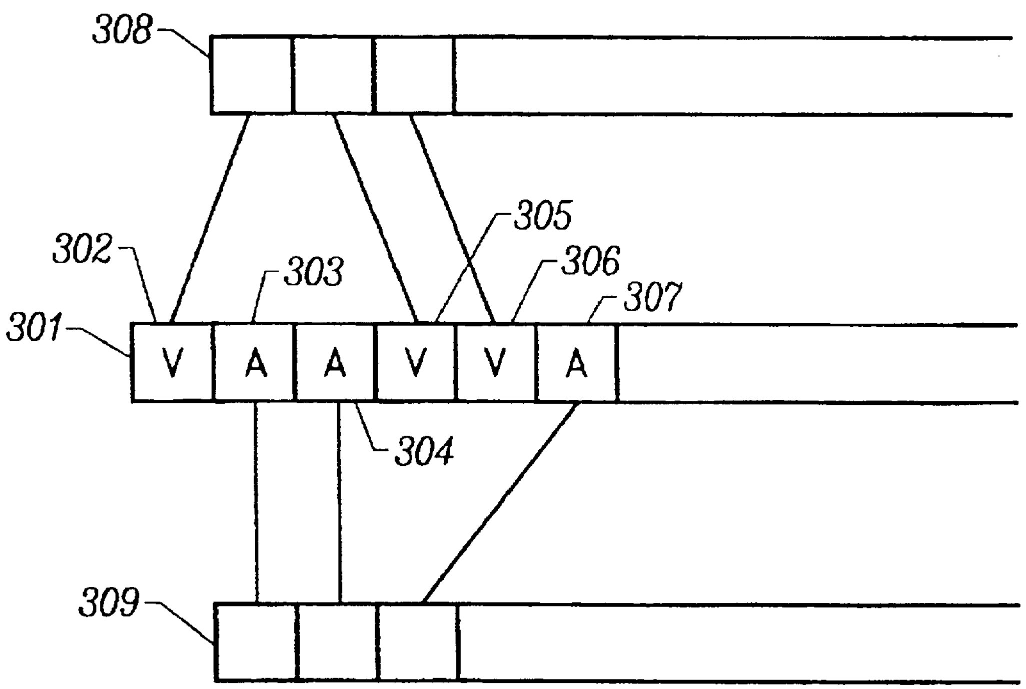

308
302
303
305
306
307
301
V
A
A
V
V
A
304
309

FIG. 3

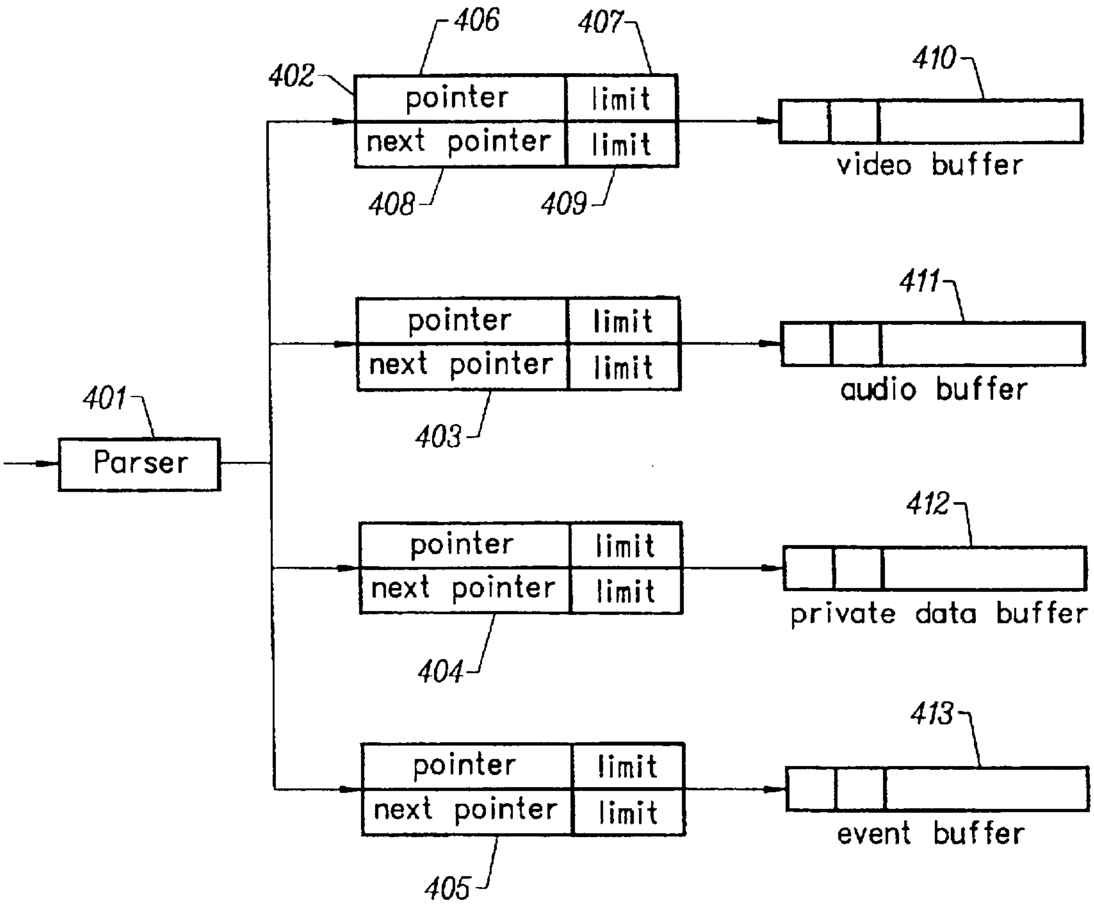

406
407
402
pointer
limit
next pointer
limit
408
409
410
video buffer
411
pointer
limit
next pointer
limit
audio buffer
403
401
Parser
412
pointer
limit
next pointer
limit
private data buffer
404
413
pointer
limit
next pointer
limit
event buffer
405

FIG. 4

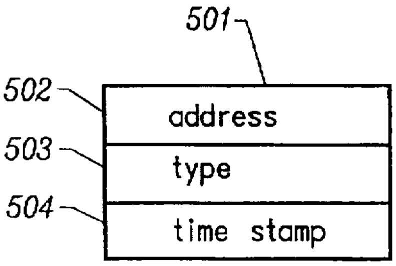

501
502
address
503
type
504
time stamp

FIG. 5

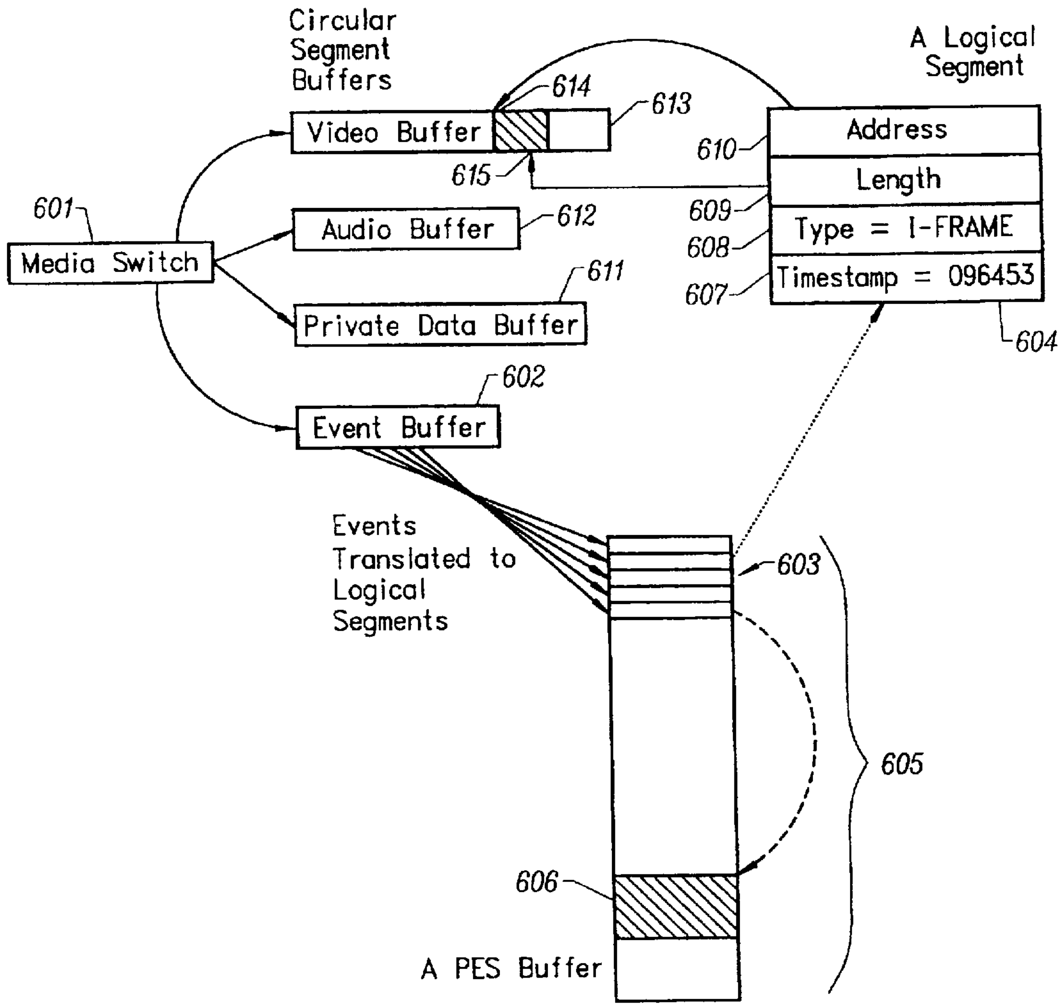

Circular
Segment
Buffers
Video Buffer
614
613
615
Audio Buffer
612
Private Data Buffer
611
601
Media Switch
Event Buffer
602
A Logical
Segment
Address
Length
Type = I-FRAME
Timestamp = 096453
610
609
608
607
604
Events
Translated to
Logical
Segments
603
605
606
A PES Buffer

FIG. 6

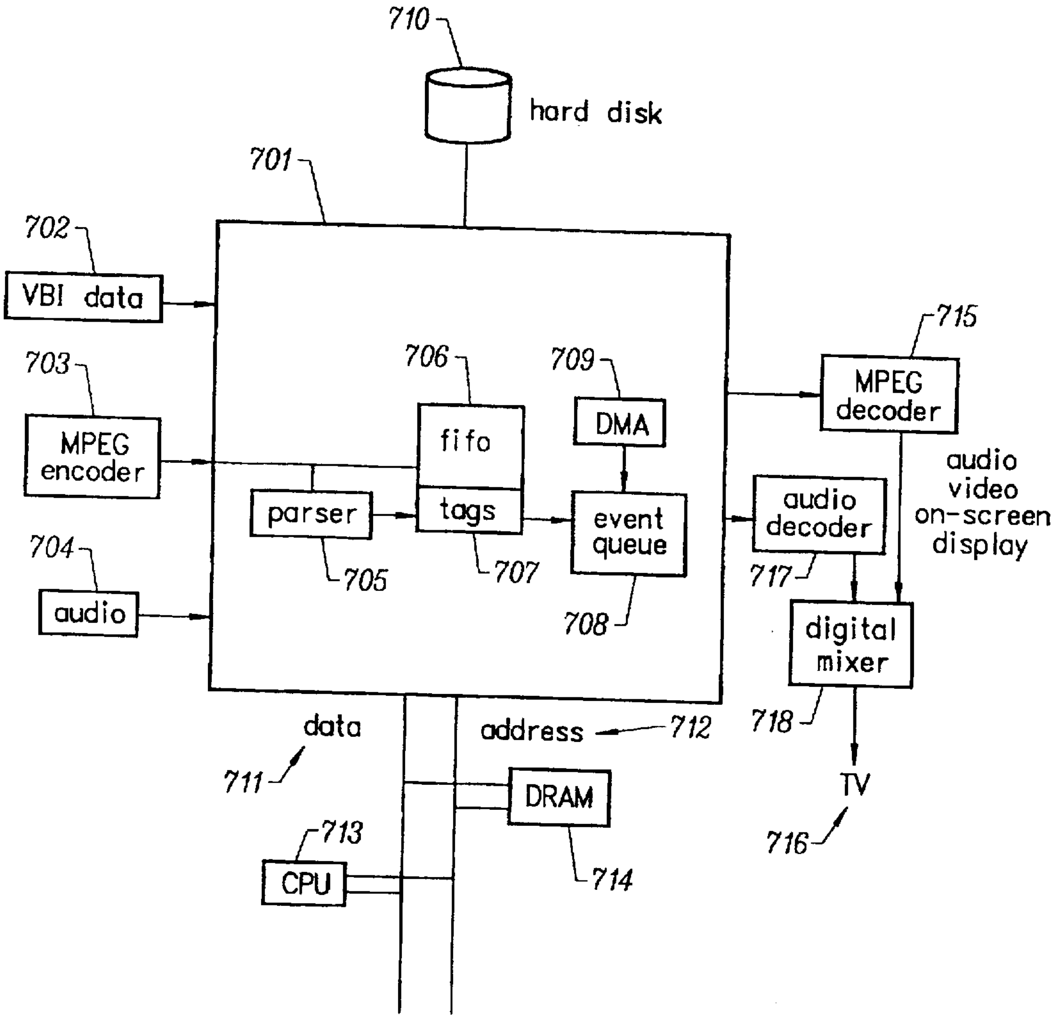

710
hard disk
701
702
VBI data
703
MPEG encoder
704
audio
706
fifo
tags
parser
705
707
709
DMA
event queue
708
715
MPEG decoder
audio video on-screen display
audio decoder
717
digital mixer
718
TV
716
data
711
address
712
DRAM
714
713
CPU

FIG. 7

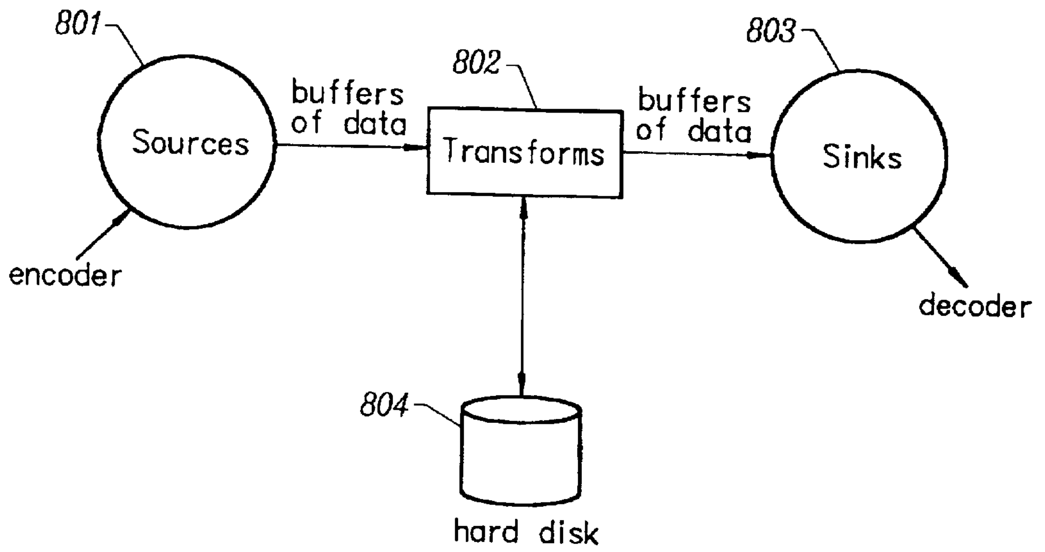
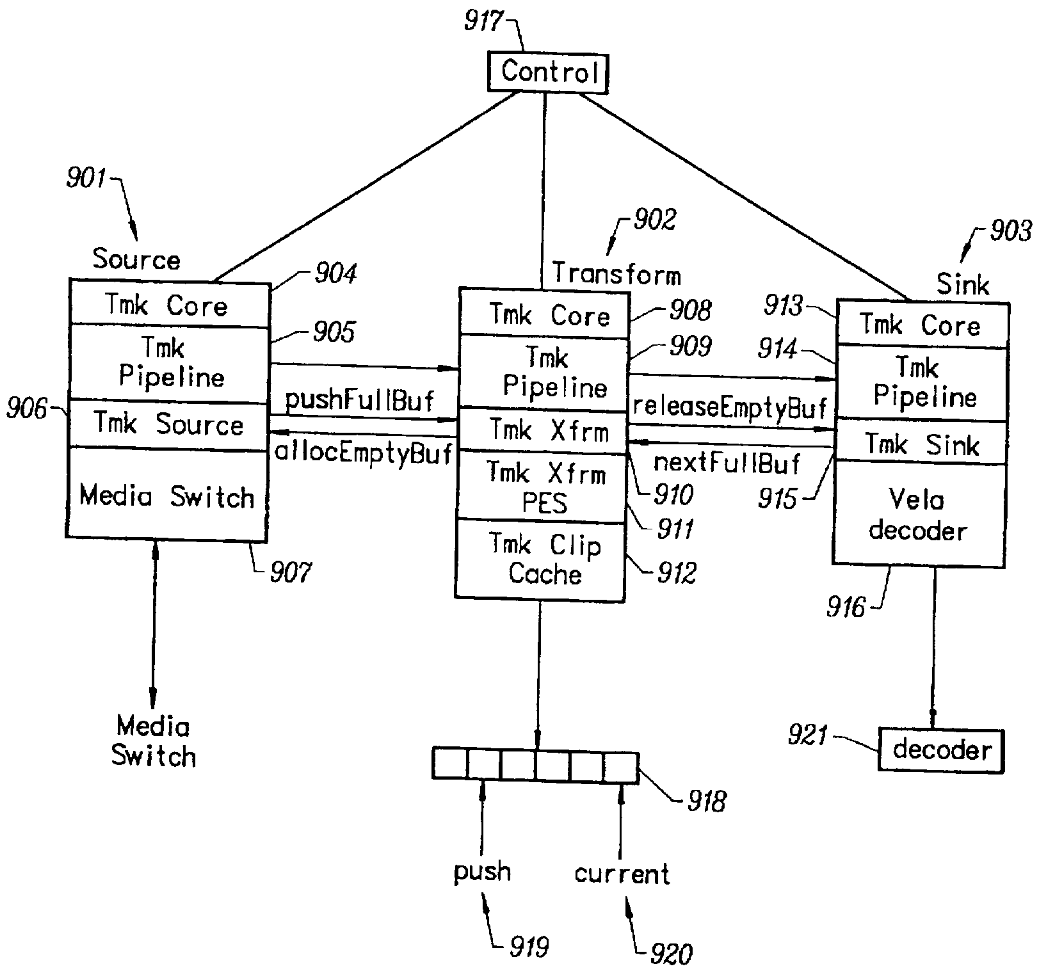

801
Sources
encoder
buffers of data
802
Transforms
buffers of data
803
Sinks
decoder
804
hard disk

FIG. 8



917
Control
901
Source
904
Tmk Core
905
Tmk Pipeline
906
Tmk Source
Media Switch
907
Media Switch
902
Transform
Tmk Core
908
Tmk Pipeline
909
Tmk Xfrm
910
Tmk Xfrm PES
911
Tmk Clip Cache
912
pushFullBuf
allocEmptyBuf
releaseEmptyBuf
nextFullBuf
903
Sink
913
Tmk Core
914
Tmk Pipeline
Tmk Sink
915
Vela decoder
916
921
decoder
918
push
current
919
920

FIG. 9

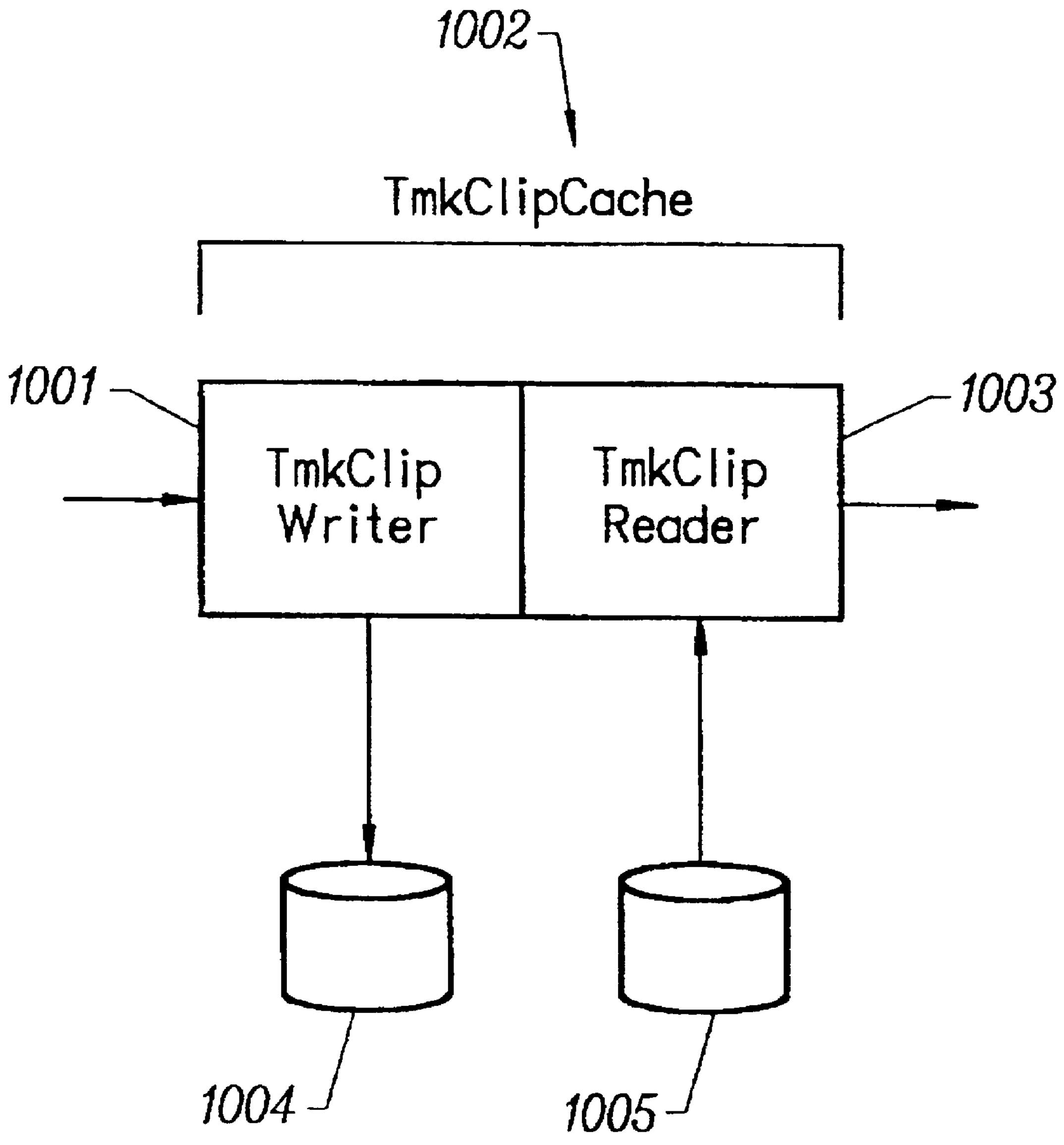

1002
TmkClipCache
1001
1003
TmkClip
Writer
TmkClip
Reader
1004
1005

FIG. 10

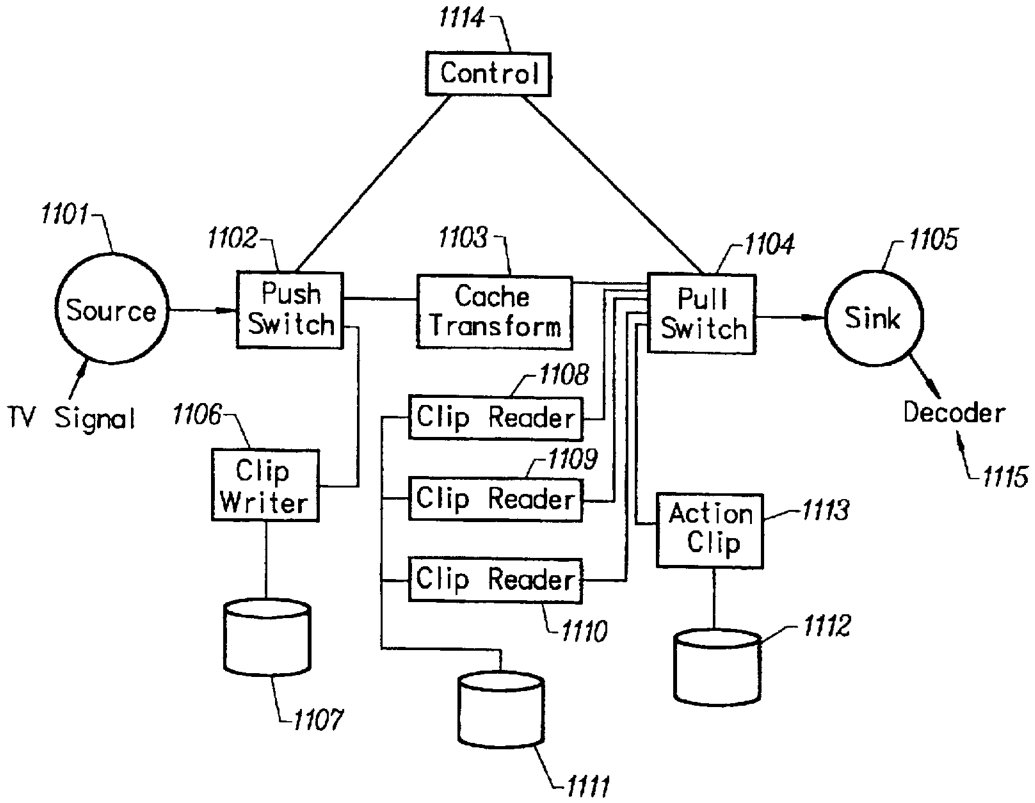

1114
Control
1101
Source
TV Signal
1102
Push Switch
1103
Cache Transform
1104
Pull Switch
1105
Sink
Decoder
1115
1106
Clip Writer
1107
1108
Clip Reader
1109
Clip Reader
Clip Reader
1110
1111
Action Clip
1113
1112

FIG. 11

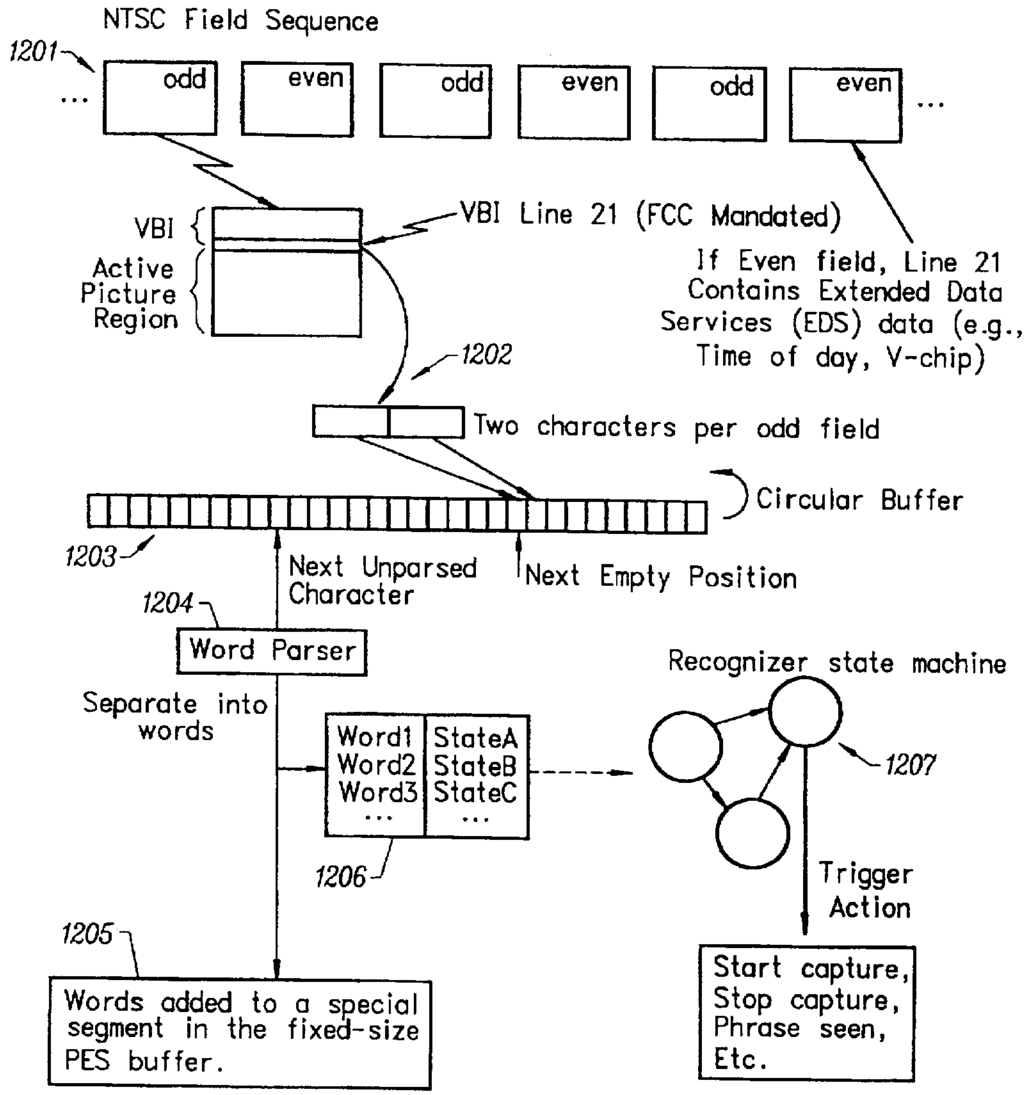

NTSC Field Sequence
1201
odd
even
odd
even
odd
even
VBI
Active Picture Region
VBI Line 21 (FCC Mandated)
If Even field, Line 21 Contains Extended Data Services (EDS) data (e.g., Time of day, V-chip)
1202
Two characters per odd field
Circular Buffer
1203
Next Unparsed Character
Next Empty Position
1204
Word Parser
Separate into words
Word1 StateA
Word2 StateB
Word3 StateC
1206
Recognizer state machine
1207
Trigger Action
1205
Words added to a special segment in the fixed-size PES buffer.
Start capture, Stop capture, Phrase seen, Etc.

FIG. 12

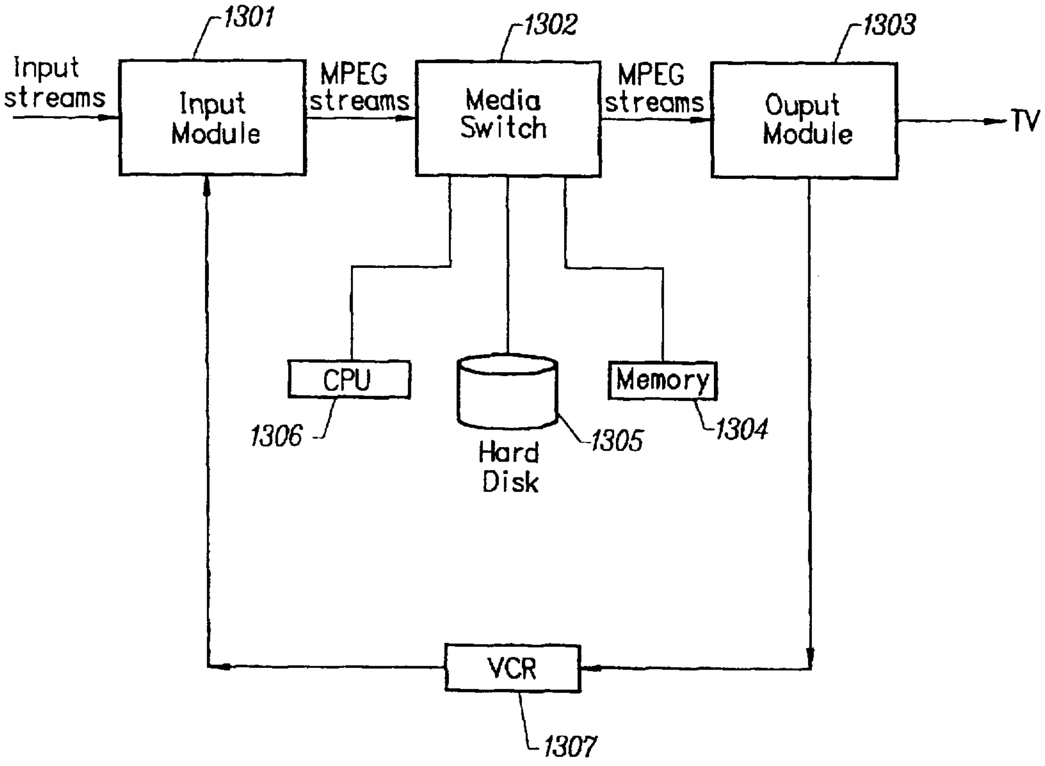

1301
Input streams
Input Module
MPEG streams
1302
Media Switch
MPEG streams
1303
Ouput Module
TV
CPU
1306
Hard Disk
1305
Memory
1304
VCR
1307

FIG. 13

MULTIMEDIA TIME WARPING SYSTEM

BACKGROUND OF THE INVENTION

1. Technical Field

The invention relates to the time shifting of television broadcast signals. More particularly, the invention relates to the real time capture, storage, and display of television broadcast signals.

2. Description of the Prior Art

The Video Cassette Recorder (VCR) has changed the lives of television (TV) viewers throughout the world. The VCR has offered viewers the flexibility to time-shift TV programs to match their lifestyles.

The viewer stores TV programs onto magnetic tape using the VCR. The VCR gives the viewer the ability to play, rewind, fast forward and pause the stored program material. These functions enable the viewer to pause the program playback whenever he desires, fast forward through unwanted program material or commercials, and to replay favorite scenes. However, a VCR cannot both capture and play back information at the same time.

One approach to solving this problem is to use several VCRs. For example, if two video tape recorders are available, it might be possible to Ping-Pong between the two. In this case, the first recorder is started at the beginning of the program of interest. If the viewer wishes to rewind the broadcast, the second recorder begins recording, while the first recorder is halted, rewound to the appropriate place, and playback initiated. However, at least a third video tape recorder is required if the viewer wishes to fast forward to some point in time after the initial rewind was requested. In this case, the third recorder starts recording the broadcast stream while the second is halted and rewound to the appropriate position. Continuing this exercise, one can quickly see that the equipment becomes unwieldy, unreliable, expensive, and hard to operate, while never supporting all desired functions. In addition, tapes are of finite length, and may potentially end at inconvenient times, drastically lowering the value of the solution.

The use of digital computer systems to solve this problem has been suggested. U.S. Pat. No. 5,371,551 issued to Logan et al., on Dec. 6, 1994, teaches a method for concurrent video recording and playback. It presents a microprocessor controlled broadcast and playback device. Said device compresses and stores video data onto a hard disk. However, this approach is difficult to implement because the processor requirements for keeping up with the high video rates makes the device expensive and problematic. The microprocessor must be extremely fast to keep up with the incoming and outgoing video data.

It would be advantageous to provide a multimedia time warping system that gives the user the ability to simultaneously record and play back TV broadcast programs. It would further be advantageous to provide a multimedia time warping system that utilizes an approach that decouples the microprocessor from the high video data rates, thereby reducing the microprocessor and system requirements which are at a premium.

SUMMARY OF THE INVENTION

The invention provides a multimedia time warping system. The invention utilizes an easily manipulated, low cost multimedia storage and display system that allows the user to view a television broadcast program with the option of instantly reviewing previous scenes within the program. In addition, the invention allows the user to store selected television broadcast programs while the user is simultaneously watching or reviewing another program.

A preferred embodiment of the invention accepts television (TV) input streams in a multitude of forms, for example, analog forms such as National Television Standards Committee (NTSC) or PAL broadcast, and digital forms such as Digital Satellite System (DSS), Digital Broadcast Services (DBS), or Advanced Television Standards Committee (ATSC). Analog TV streams are converted to an Moving Pictures Experts Group (MPEG) formatted stream for internal transfer and manipulation, while pre-formatted MPEG streams are extracted from the digital TV signal and presented in a similar format to encoded analog streams.

The invention parses the resulting MPEG stream and separates it into its video and audio components. It then stores the components into temporary buffers. Events are recorded that indicate the type of component that has been found, where it is located, and when it occurred. The program logic is notified that an event has occurred and the data is extracted from the buffers.

The parser and event buffer decouple the CPU from having to parse the MPEG stream and from the real time nature of the data streams. This decoupling allows for slower CPU and bus speeds which translate to lower system costs. The video and audio components are stored on a storage device. When the program is requested for display, the video and audio components are extracted from the storage device and reassembled into an MPEG stream. The MPEG stream is sent to a decoder. The decoder converts the MPEG stream into TV output signals and delivers the TV output signals to a TV receiver.

User control commands are accepted and sent through the system. These commands affect the flow of said MPEG stream and allow the user to view stored programs with at least the following functions: reverse, fast forward, play, pause, index, fast/slow reverse play, and fast/slow play.

Other aspects and advantages of the invention will become apparent from the following detailed description in combination with the accompanying drawings, illustrating, by way of example, the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block schematic diagram of a high level view of a preferred embodiment of the invention according to the invention;

FIG. **2** is a block schematic diagram of a preferred embodiment of the invention using multiple input and output modules according to the invention;

FIG. **3** is a schematic diagram of an Moving Pictures Experts Group (MPEG) data stream and its video and audio components according to the invention;

FIG. **4** is a block schematic diagram of a parser and four direct memory access (DMA) input engines contained in the Media Switch according to the invention;

FIG. **5** is a schematic diagram of the components of a packetized elementary stream (PES) buffer according to the invention;

FIG. **6** is a schematic diagram of the construction of a PES buffer from the parsed components in the Media Switch output circular buffers;

FIG. **7** is a block schematic diagram of the Media Switch and the various components that it communicates with according to the invention;

FIG. **8** is a block schematic diagram of a high level view of the program logic according to the invention;

FIG. **9** is a block schematic diagram of a class hierarchy of the program logic according to the invention;

FIG. **10** is a block schematic diagram of a preferred embodiment of the clip cache component of the invention according to the invention;

FIG. **11** is a block schematic diagram of a preferred embodiment of the invention that emulates a broadcast studio video mixer according to the invention;

FIG. **12** is a block schematic diagram of a closed caption parser according to the invention; and

FIG. **13** is a block schematic diagram of a high level view of a preferred embodiment of the invention utilizing a VCR as an integral component of the invention according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

The invention is embodied in a multimedia time warping system. A system according to the invention provides a multimedia storage and display system that allows the user to view a television broadcast program with the option of instantly reviewing previous scenes within the program. The invention additionally provides the user with the ability to store selected television broadcast programs while simultaneously watching or reviewing another program and to view stored programs with at least the following functions: reverse, fast forward, play, pause, index, fast/slow reverse play, and fast/slow play.

Referring to FIG. 1, a preferred embodiment of the invention has an Input Section **101**, Media Switch **102**, and an Output Section **103**. The Input Section **101** takes television (TV) input streams in a multitude of forms, for example, National Television Standards Committee (NTSC) or PAL broadcast, and digital forms such as Digital Satellite System (DSS), Digital Broadcast Services (DBS), or Advanced Television Standards Committee (ATSC). DBS, DSS and ATSC are based on standards called Moving Pictures Experts Group 2 (MPEG2) and MPEG2 Transport. MPEG2 Transport is a standard for formatting the digital data stream from the TV source transmitter so that a TV receiver can disassemble the input stream to find programs in the multiplexed signal. The Input Section **101** produces MPEG streams. An MPEG2 transport multiplex supports multiple programs in the same broadcast channel, with multiple video and audio feeds and private data. The Input Section **101** tunes the channel to a particular program, extracts a specific MPEG program out of it, and feeds it to the rest of the system. Analog TV signals are encoded into a similar MPEG format using separate video and audio encoders, such that the remainder of the system is unaware of how the signal was obtained. Information may be modulated into the Vertical Blanking Interval (VBI) of the analog TV signal in a number of standard ways; for example, the North American Broadcast Teletext Standard (NABTS) may be used to modulate information onto lines **10** through **20** of an NTSC signal, while the FCC mandates the use of line **21** for Closed Caption (CC) and Extended Data Services (EDS). Such signals are decoded by the input section and passed to the other sections as if they were delivered via an MPEG2 private data channel.

The Media Switch **102** mediates between a microprocessor CPU **106**, hard disk or storage device **105**, and memory **104**. Input streams are converted to an MPEG stream and sent to the Media Switch **102**. The Media Switch **102** buffers the MPEG stream into memory. It then performs two operations if the user is watching real time TV: the stream is sent to the Output Section **103** and it is written simultaneously to the hard disk or storage device **105**.

The Output Section **103** takes MPEG streams as input and produces an analog TV signal according to the NTSC, PAL, or other required TV standards. The Output Section **103** contains an MPEG decoder, On-Screen Display (OSD) generator, analog TV encoder and audio logic. The OSD generator allows the program logic to supply images which will be overlayed on top of the resulting analog TV signal. Additionally, the Output Section can modulate information supplied by the program logic onto the VBI of the output signal in a number of standard formats, including NABTS, CC and EDS.

With respect to FIG. **2**, the invention easily expands to accommodate multiple Input Sections (tuners) **201**, **202**, **203**, **204**, each can be tuned to different types of input. Multiple Output Modules (decoders) **206**, **207**, **208**, **209** are added as well. Special effects such as picture in a picture can be implemented with multiple decoders. The Media Switch **205** records one program while the user is watching another. This means that a stream can be extracted off the disk while another stream is being stored onto the disk.

Referring to FIG. **3**, the incoming MPEG stream **301** has interleaved video **302**, **305**, **306** and audio **303**, **304**, **307** segments. These elements must be separated and recombined to create separate video **308** and audio **309** streams or buffers. This is necessary because separate decoders are used to convert MPEG elements back into audio or video analog components. Such separate delivery requires that time sequence information be generated so that the decoders may be properly synchronized for accurate playback of the signal.

The Media Switch enables the program logic to associate proper time sequence information with each segment, possibly embedding it directly into the stream. The time sequence information for each segment is called a time stamp. These time stamps are monotonically increasing and start at zero each time the system boots up. This allows the invention to find any particular spot in any particular video segment. For example, if the system needs to read five seconds into an incoming contiguous video stream that is being cached, the system simply has to start reading forward into the stream and look for the appropriate time stamp.

A binary search can be performed on a stored file to index into a stream. Each stream is stored as a sequence of fixed-size segments enabling fast binary searches because of the uniform time stamping. If the user wants to start in the middle of the program, the system performs a binary search of the stored segments until it finds the appropriate spot, obtaining the desired results with a minimal amount of information. If the signal were instead stored as an MPEG stream, it would be necessary to linearly parse the stream from the beginning to find the desired location.

With respect to FIG. **4**, the Media Switch contains four input Direct Memory Access (DMA) engines **402**, **403**, **404**, **405** each DMA engine has an associated buffer **410**, **411**, **412**, **413**. Conceptually, each DMA engine has a pointer **406**, a limit for that pointer **407**, a next pointer **408**, and a limit for the next pointer **409**. Each DMA engine is dedicated to a particular type of information, for example, video **402**, audio **403**, and parsed events **405**. The buffers **410**, **411**, **412**, **413** are circular and collect the specific information. The DMA engine increments the pointer **406** into the associated buffer until it reaches the limit **407** and then loads the next pointer **408** and limit **409**. Setting the pointer **406** and next pointer **408** to the same value, along with the corresponding limit value creates a circular buffer. The next pointer **408** can be set to a different address to provide vector DMA.

The input stream flows through a parser **401**. The parser **401** parses the stream looking for MPEG distinguished events indicating the start of video, audio or private data segments. For example, when the parser **401** finds a video event, it directs the stream to the video DMA engine **402**. The parser **401** buffers up data and DMAs it into the video buffer **410** through the video DMA engine **402**. At the same time, the parser **401** directs an event to the event DMA engine **405** which generates an event into the event buffer **413**. When the parser **401** sees an audio event, it redirects the byte stream to the audio DMA engine **403** and generates an event into the event buffer **413**. Similarly, when the parser **401** sees a private data event, it directs the byte stream to the private data DMA engine **404** and directs an event to the event buffer **413**. The Media Switch notifies the program logic via an interrupt mechanism when events are placed in the event buffer.

Referring to FIGS. **4** and **5**, the event buffer **413** is filled by the parser **401** with events. Each event **501** in the event buffer has an offset **502**, event type **503**, and time stamp field **504**. The parser **401** provides the type and offset of each event as it is placed into the buffer. For example, when an audio event occurs, the event type field is set to an audio event and the offset indicates the location in the audio buffer **411**. The program logic knows where the audio buffer **411** starts and adds the offset to find the event in the stream. The address offset **502** tells the program logic where the next event occurred, but not where it ended. The previous event is cached so the end of the current event can be found as well as the length of the segment.

With respect to FIGS. **5** and **6**, the program logic reads accumulated events in the event buffer **602** when it is interrupted by the Media Switch **601**. From these events the program logic generates a sequence of logical segments **603** which correspond to the parsed MPEG segments **615**. The program logic converts the offset **502** into the actual address **610** of each segment, and records the event length **609** using the last cached event. If the stream was produced by encoding an analog signal, it will not contain Program Time Stamp (PTS) values, which are used by the decoders to properly present the resulting output. Thus, the program logic uses the generated time stamp **504** to calculate a simulated PTS for each segment and places that into the logical segment time stamp **607**. In the case of a digital TV stream, PTS values are already encoded in the stream. The program logic extracts this information and places it in the logical segment time stamp **607**.

The program logic continues collecting logical segments **603** until it reaches the fixed buffer size. When this occurs, the program logic generates a new buffer, called a Packetized Elementary Stream (PES) **605** buffer containing these logical segments **603** in order, plus ancillary control information. Each logical segment points **604** directly to the circular buffer, e.g., the video buffer **613**, filled by the Media Switch **601**. This new buffer is then passed to other logic components, which may further process the stream in the buffer in some way, such as presenting it for decoding or writing it to the storage media. Thus, the MPEG data is not copied from one location in memory to another by the processor. This results in a more cost effective design since lower memory bandwidth and processor bandwidth is required.

A unique feature of the MPEG stream transformation into PES buffers is that the data associated with logical segments need not be present in the buffer itself, as presented above. When a PES buffer is written to storage, these logical segments are written to the storage medium in the logical order in which they appear. This has the effect of gathering components of the stream, whether they be in the video, audio or private data circular buffers, into a single linear buffer of stream data on the storage medium. The buffer is read back from the storage medium with a single transfer from the storage media, and the logical segment information is updated to correspond with the actual locations in the buffer **606**. Higher level program logic is unaware of this transformation, since it handles only the logical segments, thus stream data is easily managed without requiring that the data ever be copied between locations in DRAM by the CPU.

A unique aspect of the Media Switch is the ability to handle high data rates effectively and inexpensively. It performs the functions of taking video and audio data in, sending video and audio data out, sending video and audio data to disk, and extracting video and audio data from the disk on a low cost platform. Generally, the Media Switch runs asynchronously and autonomously with the microprocessor CPU, using its DMA capabilities to move large quantities of information with minimal intervention by the CPU.

Referring to FIG. **7**, the input side of the Media Switch **701** is connected to an MPEG encoder **703**. There are also circuits specific to MPEG audio **704** and vertical blanking interval (VBI) data **702** feeding into the Media Switch **701**. If a digital TV signal is being processed instead, the MPEG encoder **703** is replaced with an MPEG2 Transport Demultiplexor, and the MPEG audio encoder **704** and VBI decoder **702** are deleted. The demultiplexor multiplexes the extracted audio, video and private data channel streams through the video input Media Switch port.

The parser **705** parses the input data stream from the MPEG encoder **703**, audio encoder **704** and VBI decoder **702**, or from the transport demultiplexor in the case of a digital TV stream. The parser **705** detects the beginning of all of the important events in a video or audio stream, the start of all of the frames, the start of sequence headers—all of the pieces of information that the program logic needs to know about in order to both properly play back and perform special effects on the stream, e.g. fast forward, reverse, play, pause, fast/slow play, indexing, and fast/slow reverse play.

The parser **705** places tags **707** into the FIFO **706** when it identifies video or audio segments, or is given private data. The DMA **709** controls when these tags are taken out. The tags **707** and the DMA addresses of the segments are placed into the event queue **708**. The frame type information, whether it is a start of a video I-frame, video B-frame, video P-frame, video PES, audio PES, a sequence header, an audio frame, or private data packet, is placed into the event queue **708** along with the offset in the related circular buffer where the piece of information was placed. The program logic operating in the CPU **713** examines events in the circular buffer after it is transferred to the DRAM **714**.

The Media Switch **701** has a data bus **711** that connects to the CPU **713** and DRAM **714**. An address bus **712** is also shared between the Media Switch **701**, CPU **713**, and DRAM **714**. A hard disk or storage device **710** is connected to one of the ports of the Media Switch **701**. The Media Switch **701** outputs streams to an MPEG video decoder **715** and a separate audio decoder **717**. The audio decoder **717** signals contain audio cues generated by the system in response to the user's commands on a remote control or other internal events. The decoded audio output from the MPEG decoder is digitally mixed 718 with the separate audio signal. The resulting signals contain video, audio, and on-screen displays and are sent to the TV 716.

The Media Switch 701 takes in 8-bit data and sends it to the disk, while at the same time extracts another stream of data off of the disk and sends it to the MPEG decoder 715. All of the DMA engines described above can be working at the same time. The Media Switch 701 can be implemented in hardware using a Field Programmable Gate Array (FPGA), ASIC, or discrete logic.

Rather than having to parse through an immense data stream looking for the start of where each frame would be, the program logic only has to look at the circular event buffer in DRAM 714 and it can tell where the start of each frame is and the frame type. This approach saves a large amount of CPU power, keeping the real time requirements of the CPU 713 small. The CPU 713 does not have to be very fast at any point in time. The Media Switch 701 gives the CPU 713 as much time as possible to complete tasks. The parsing mechanism 705 and event queue 708 decouple the CPU 713 from parsing the audio, video, and buffers and the real time nature of the streams, which allows for lower costs. It also allows the use of a bus structure in a CPU environment that operates at a much lower clock rate with much cheaper memory than would be required otherwise.

The CPU 713 has the ability to queue up one DMA transfer and can set up the next DMA transfer at its leisure. This gives the CPU 713 large time intervals within which it can service the DMA controller 709. The CPU 713 may respond to a DMA interrupt within a larger time window because of the large latency allowed. MPEG streams, whether extracted from an MPEG2 Transport or encoded from an analog TV signal, are typically encoded using a technique called Variable Bit Rate encoding (VBR). This technique varies the amount of data required to represent a sequence of images by the amount of movement between those images. This technique can greatly reduce the required bandwidth for a signal, however sequences with rapid movement (such as a basketball game) may be encoded with much greater bandwidth requirements. For example, the Hughes DirecTV satellite system encodes signals with anywhere from 1 to 10 Mb/s of required bandwidth, varying from frame to frame. It would be difficult for any computer system to keep up with such rapidly varying data rates without this structure.

With respect to FIG. 8, the program logic within the CPU has three conceptual components: sources 801, transforms 802, and sinks 803. The sources 801 produce buffers of data. Transforms 802 process buffers of data and sinks 803 consume buffers of data. A transform is responsible for allocating and queuing the buffers of data on which it will operate. Buffers are allocated as if "empty" to sources of data, which give them back "full". The buffers are then queued and given to sinks as "full", and the sink will return the buffer "empty".

A source 801 accepts data from encoders, e.g., a digital satellite receiver. It acquires buffers for this data from the downstream transform, packages the data into a buffer, then pushes the buffer down the pipeline as described above. The source object 801 does not know anything about the rest of the system. The sink 803 consumes buffers, taking a buffer from the upstream transform, sending the data to the decoder, and then releasing the buffer for reuse.

There are two types of transforms 802 used: spatial and temporal. Spatial transforms are transforms that perform, for example, an image convolution or compression/decompression on the buffered data that is passing through. Temporal transforms are used when there is no time relation that is expressible between buffers going in and buffers coming out of a system. Such a transform writes the buffer to a file 804 on the storage medium. The buffer is pulled out at a later time, sent down the pipeline, and properly sequenced within the stream.

Referring to FIG. 9, a C++ class hierarchy derivation of the program logic is shown. The TiVo Media Kernel (Tmk) 904, 908, 913 mediates with the operating system kernel. The kernel provides operations such as: memory allocation, synchronization, and threading. The TmkCore 904, 908, 913 structures memory taken from the media kernel as an object. It provides operators, new and delete, for constructing and deconstructing the object. Each object (source 901, transform 902, and sink 903) is multi-threaded by definition and can run in parallel.

The TmkPipeline class 905, 909, 914 is responsible for flow control through the system. The pipelines point to the next pipeline in the flow from source 901 to sink 903. To pause the pipeline, for example, an event called "pause" is sent to the first object in the pipeline. The event is relayed on to the next object and so on down the pipeline. This all happens asynchronously to the data going through the pipeline. Thus, similar to applications such as telephony, control of the flow of MPEG streams is asynchronous and separate from the streams themselves. This allows for a simple logic design that is at the same time powerful enough to support the features described previously, including pause, rewind, fast forward and others. In addition, this structure allows fast and efficient switching between stream sources, since buffered data can be simply discarded and decoders reset using a single event, after which data from the new stream will pass down the pipeline. Such a capability is needed, for example, when switching the channel being captured by the input section, or when switching between a live signal from the input section and a stored stream.

The source object 901 is a TmkSource 906 and the transform object 902 is a TmkXfrm 910. These are intermediate classes that define standard behaviors for the classes in the pipeline. Conceptually, they handshake buffers down the pipeline. The source object 901 takes data out of a physical data source, such as the Media Switch, and places it into a PES buffer. To obtain the buffer, the source object 901 asks the down stream object in his pipeline for a buffer (allocEmptyBuf). The source object 901 is blocked until there is sufficient memory. This means that the pipeline is self-regulating; it has automatic flow control. When the source object 901 has filled up the buffer, it hands it back to the transform 902 through the pushFullBuf function.

The sink 903 is flow controlled as well. It calls nextFullBuf which tells the transform 902 that it is ready for the next filled buffer. This operation can block the sink 903 until a buffer is ready. When the sink 903 is finished with a buffer (i.e., it has consumed the data in the buffer) it calls releaseEmptyBuf. ReleaseEmptyBuf gives the buffer back to the transform 902. The transform 902 can then hand that buffer, for example, back to the source object 901 to fill up again. In addition to the automatic flow-control benefit of this method, it also provides for limiting the amount of memory dedicated to buffers by allowing enforcement of a fixed allocation of buffers by a transform. This is an important feature in achieving a cost-effective limited DRAM environment.

The MediaSwitch class 909 calls the allocEmptyBuf method of the TmkClipCache 912 object and receives a PES buffer from it. It then goes out to the circular buffers in the Media Switch hardware and generates PES buffers. The MediaSwitch class **909** fills the buffer up and pushes it back to the TmkClipCache **912** object.

The TmkClipCache **912** maintains a cache file **918** on a storage medium. It also maintains two pointers into this cache: a push pointer **919** that shows where the next buffer coming from the source **901** is inserted; and a current pointer **920** which points to the current buffer used.

The buffer that is pointed to by the current pointer is handed to the Vela decoder class **916**. The Vela decoder class **916** talks to the decoder **921** in the hardware. The decoder **921** produces a decoded TV signal that is subsequently encoded into an analog TV signal in NTSC, PAL or other analog format. When the Vela decoder class **916** is finished with the buffer it calls releaseEmptyBuf.

The structure of the classes makes the system easy to test and debug. Each level can be tested separately to make sure it performs in the appropriate manner, and the classes may be gradually aggregated to achieve the desired functionality while retaining the ability to effectively test each object.

The control object **917** accepts commands from the user and sends events into the pipeline to control what the pipeline is doing. For example, if the user has a remote control and is watching TV, the user presses pause and the control object **917** sends an event to the sink **903**, that tells it pause. The sink **903** stops asking for new buffers. The current pointer **920** stays where it is at. The sink **903** starts taking buffers out again when it receives another event that tells it to play. The system is in perfect synchronization; it starts from the frame that it stopped at.

The remote control may also have a fast forward key. When the fast forward key is pressed, the control object **917** sends an event to the transform **902**, that tells it to move forward two seconds. The transform **902** finds that the two second time span requires it to move forward three buffers. It then issues a reset event to the downstream pipeline, so that any queued data or state that may be present in the hardware decoders is flushed. This is a critical step, since the structure of MPEG streams requires maintenance of state across multiple frames of data, and that state will be rendered invalid by repositioning the pointer. It then moves the current pointer **920** forward three buffers. The next time the sink **903** calls nextFullBuf it gets the new current buffer. The same method works for fast reverse in that the transform **902** moves the current pointer **920** backwards.

A system clock reference resides in the decoder. The system clock reference is sped up for fast play or slowed down for slow play. The sink simply asks for full buffers faster or slower, depending on the clock speed.

With respect to FIG. **10**, two other objects derived from the TmkXfrm class are placed in the pipeline for disk access. One is called TmkClipReader **1003** and the other is called TmkClipWriter **1001**. Buffers come into the TmkClipWriter **1001** and are pushed to a file on a storage medium **1004**. TmkClipReader **1003** asks for buffers which are taken off of a file on a storage medium **1005**. A TmkClipReader **1003** provides only the allocEmptyBuf and pushFullBuf methods, while a TmkClipWriter **1001** provides only the nextFullBuf and releaseEmptyBuf methods. A TmkClipReader **1003** therefore performs the same function as the input, or "push" side of a TmkClipCache **1002**, while a TmkClipWriter **1001** therefore performs the same function as the output, or "pull" side of a TmkClipCache **1002**.

Referring to FIG. **11**, a preferred embodiment that accomplishes multiple functions is shown. A source **1101** has a TV signal input. The source sends data to a PushSwitch **1102** which is a transform derived from TmkXfrm. The PushSwitch **1102** has multiple outputs that can be switched by the control object **1114**. This means that one part of the pipeline can be stopped and another can be started at the users whim. The user can switch to different storage devices. The PushSwitch **1102** could output to a TmkClipWriter **1106**, which goes onto a storage device **1107** or write to the cache transform **1103**.

An important feature of this apparatus is the ease with which it can selectively capture portions of an incoming signal under the control of program logic. Based on information such as the current time, or perhaps a specific time span, or perhaps via a remote control button press by the viewer, a TmkClipWriter **1106** may be switched on to record a portion of the signal, and switched off at some later time. This switching is typically caused by sending a "switch" event to the PushSwitch **1102** object.

An additional method for triggering selective capture is through information modulated into the VBI or placed into an MPEG private data channel. Data decoded from the VBI or private data channel is passed to the program logic. The program logic examines this data to determine if the data indicates that capture of the TV signal into which it was modulated should begin. Similarly, this information may also indicate when recording should end, or another data item may be modulated into the signal indicating when the capture should end. The starting and ending indicators may be explicitly modulated into the signal or other information that is placed into the signal in a standard fashion may be used to encode this information.

With respect to FIG. **12**, an example is shown which demonstrates how the program logic scans the words contained within the closed caption (CC) fields to determine starting and ending times, using particular words or phrases to trigger the capture. A stream of NTSC or PAL fields **1201** is presented. CC bytes are extracted from each odd field **1202**, and entered in a circular buffer **1203** for processing by the Word Parser **1204**. The Word Parser **1204** collects characters until it encounters a word boundary, usually a space, period or other delineating character. Recall from above, that the MPEG audio and video segments are collected into a series of fixed-size PES buffers. A special segment is added to each PES buffer to hold the words extracted from the CC field **1205**. Thus, the CC information is preserved in time synchronization with the audio and video, and can be correctly presented to the viewer when the stream is displayed. This also allows the stored stream to be processed for CC information at the leisure of the program logic, which spreads out load, reducing cost and improving efficiency. In such a case, the words stored in the special segment are simply passed to the state table logic **1206**.

During stream capture, each word is looked up in a table **1206** which indicates the action to take on recognizing that word. This action may simply change the state of the recognizer state machine **1207**, or may cause the state machine **1207** to issue an action request, such as "start capture", "stop capture", "phrase seen", or other similar requests. Indeed, a recognized word or phrase may cause the pipeline to be switched; for example, to overlay a different audio track if undesirable language is used in the program.

Note that the parsing state table **1206** and recognizer state machine **1207** may be modified or changed at any time. For example, a different table and state machine may be provided for each input channel. Alternatively, these elements may be switched depending on the time of day, or because of other events.

Referring to FIG. **11**, a PullSwitch is added **1104** which outputs to the sink **1105**.

The sink **1105** calls nextFullBuf and releaseEmptyBuf to get or return buffers from the PullSwitch **1104**. The PullSwitch **1104** can have any number of inputs. One input could be an ActionClip **1113**. The remote control can switch between input sources. The control object **1114** sends an event to the PullSwitch **1104**, telling it to switch. It will switch from the current input source to whatever input source the control object selects.

An ActionClip class provides for sequencing a number of different stored signals in a predictable and controllable manner, possibly with the added control of viewer selection via a remote control. Thus, it appears as a derivative of a TmkXfrm object that accepts a "switch" event for switching to the next stored signal.

This allows the program logic or user to create custom sequences of video output. Any number of video segments can be lined up and combined as if the program logic or user were using a broadcast studio video mixer. TmkClipReaders **1108, 1109, 1110** are allocated and each is hooked into the PullSwitch **1104**. The PullSwitch **1104** switches between the TmkClipReaders **1108, 1109, 1110** to combine video and audio clips. Flow control is automatic because of the way the pipeline is constructed. The Push and Pull Switches are the same as video switches in a broadcast studio.

The derived class and resulting objects described here may be combined in an arbitrary way to create a number of different useful configurations for storing, retrieving, switching and viewing of TV streams. For example, if multiple input and output sections are available, one input is viewed while another is stored, and a picture-in-picture window generated by the second output is used to preview previously stored streams. Such configurations represent a unique and novel application of software transformations to achieve the functionality expected of expensive, sophisticated hardware solutions within a single cost-effective device.

With respect to FIG. **13**, a high-level system view is shown which implements a VCR backup. The Output Module **1303** sends TV signals to the VCR **1307**. This allows the user to record TV programs directly on to video tape. The invention allows the user to queue up programs from disk to be recorded on to video tape and to schedule the time that the programs are sent to the VCR **1307**. Title pages (EPG data) can be sent to the VCR **1307** before a program is sent. Longer programs can be scaled to fit onto smaller video tapes by speeding up the play speed or dropping frames.

The VCR **1307** output can also be routed back into the Input Module **1301**. In this configuration the VCR acts as a backup system for the Media Switch **1302**. Any overflow storage or lower priority programming is sent to the VCR **1307** for later retrieval.

The Input Module **1301** can decode and pass to the remainder of the system information encoded on the Vertical Blanking Interval (VBI). The Output Module **1303** can encode into the output VBI data provided by the remainder of the system. The program logic may arrange to encode identifying information of various kinds into the output signal, which will be recorded onto tape using the VCR **1307**. Playing this tape back into the input allows the program logic to read back this identifying information, such that the TV signal recorded on the tape is properly handled. For example, a particular program may be recorded to tape along with information about when it was recorded, the source network, etc. When this program is played back into the Input Module, this information can be used to control storage of the signal, presentation to the viewer, etc.

One skilled in the art will readily appreciate that such a mechanism may be used to introduce various data items to the program logic which are not properly conceived of as television signals. For instance, software updates or other data may be passed to the system. The program logic receiving this data from the television stream may impose controls on how the data is handled, such as requiring certain authentication sequences and/or decrypting the embedded information according to some previously acquired key. Such a method works for normal broadcast signals as well, leading to an efficient means of providing non-TV control information and data to the program logic.

Additionally, one skilled in the art will readily appreciate that although a VCR is specifically mentioned above, any multimedia recording device (e.g., a Digital Video Disk-Random Access Memory (DVD-RAM) recorder) is easily substituted in its place.

Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention. For example, the invention can be used in the detection of gambling casino crime. The input section of the invention is connected to the casino's video surveillance system. Recorded video is cached and simultaneously output to external VCRs. The user can switch to any video feed and examine (i.e., rewind, play, slow play, fast forward, etc.) a specific segment of the recorded video while the external VCRs are being loaded with the real-time input video. Accordingly, the invention should only be limited by the claims included below.

What is claimed is:

1. A process for the simultaneous storage and play back of multimedia data, comprising the steps of:

accepting television (TV) broadcast signals, wherein said TV signals are based on a multitude of standards, including, but not limited to, National Television Standards Committee (NTSC) broadcast, PAL broadcast, satellite transmission, DSS, DBS, or ATSC;

tuning said TV signals to a specific program;

providing at least one Input Section, wherein said Input Section converts said specific program to an Moving Pictures Experts Group (MPEG) formatted stream for internal transfer and manipulation;

providing a Media Switch, wherein said Media Switch parses said MPEG stream, said MPEG stream is separated into its video and audio components;

storing said video and audio components on a storage device;

providing at least one Output Section, wherein said Output Section extracts said video and audio components from said storage device;

wherein said Output Section assembles said video and audio components into an MPEG stream;

wherein said Output Section sends said MPEG stream to a decoder;

wherein said decoder converts said MPEG stream into TV output signals;

wherein said decoder delivers said TV output signals to a TV receiver; and

accepting control commands from a user, wherein said control commands are sent through the system and affect the flow of said MPEG stream.

2. The process of claim 1, wherein said Input Section directs said MPEG stream to the destination indicated by said control commands.

3. The process of claim 1, wherein said Output Section extracts said video and audio components from the storage device indicated by said control commands.

4. The process of claim 1, further comprising the step of:

creating custom video output sequences, wherein said sequences are specified by a user or program control.

5. The process of claim 1, wherein the storing and extracting of said video and audio components from said storage device are performed simultaneously.

6. The process of claim 1, wherein said Media Switch calculates and logically associates a time stamp to said video and audio components.

7. The process of claim 1, wherein said Media Switch extracts time stamp values from a digital TV stream and logically associates said time stamp values to said video and audio components.

8. The process of claim 1, further comprising the steps of:

placing said video component into a circular video buffer;

posting an event in a circular event buffer, wherein said event contains an indication that a video component was found and the location of said video component in said circular video buffer; and

sending notice of said event posting.

9. The process of claim 1, further comprising the steps of:

placing said audio component into a circular audio buffer;

posting an event in a circular event buffer, wherein said event contains an indication that an audio component was found and the location of said audio component in said circular audio buffer; and

sending notice of said event posting.

10. The process of claims 8 or 9, further comprising the steps of:

receiving said notice;

retrieving said event posting from said event buffer; and

indexing into the appropriate buffer indicated by the type and location information in said event buffer.

11. The process of claim 10, further comprising the steps of:

generating a buffer containing the logical audio or video segments in order, including ancillary information, wherein each of said logical segments points to the appropriate circular buffer location where corresponding audio or video components have been placed.

12. The process of claim 1, further comprising the step of:

increasing the decoder system clock rate for fast playback or fast reverse playback.

13. The process of claim 1, further comprising the step of:

decreasing the decoder system clock rate for slow playback or slow reverse playback.

14. The process of claim 1, further comprising the step of:

combining system audio cues and on-screen displays with said TV output signals.

15. The process of claim 1, further comprising the steps of:

decoding the Vertical Blanking Interval (VBI) data or private data channel information from said TV signal; and

examining said data to determine the starting or ending indicators of a specific program.

16. The process of claim 1, further comprising the step of:

scanning the words contained within the closed caption (CC) fields to determine program starting and ending times, wherein particular words or phrases are used to trigger the recording of a specific program and wherein the CC information is preserved in time synchronization with the audio and video, and can be correctly presented to the viewer when the stream is displayed.

17. The process of claim 16, further comprising the step of:

performing a specific action when a specific word is found in said CC information.

18. The process of claim 1, wherein said Media Switch has a data bus connecting it to a CPU and DRAM.

19. The process of claim 1, wherein said Media Switch shares an address bus with a CPU and DRAM.

20. The process of claim 1, wherein said Media Switch operates asychronously and autonomously with a CPU.

21. The process of claim 1, wherein said storage device is connected to said Media Switch.

22. The process of claim 1, wherein said Media Switch allows the CPU to queue up Direct Memory Access (DMA) transfers.

23. The process of claim 1, wherein said Media Switch is implemented in hardware.

24. The process of claim 1, further comprising the step of:

providing a multimedia recording device, including, but not limited to, a Video Cassette Recorder (VCR) or a Digital Video Disk-Random Access Memory (DVD-RAM) device, wherein said recording device is attached to the output side of said decoder, allowing said user to record said TV output signals.

25. The process of claim 24, wherein said user queues up programs from said storage device to be stored on said recording device.

26. The process of claim 24, wherein said user sets time schedules for said programs to be sent to said recording device.

27. The process of claim 24, wherein title pages may be sent to said recording device before sending a program to be stored on said recording device.

28. The process of claim 24, wherein a program that is longer in duration than a magnetic tape in said recording device allows, is sped up to fit within the desired time limit.

29. The process of claim 24, wherein a program that is longer in duration than a magnetic tape in said recording device allows, has frames dropped from it to fit within the desired time limit.

30. The process of claim 24, wherein the output of said recording device is routed to said Input Section, allowing said recording device to act as a storage back up system, said recording device accepts overflow storage, TV programs, software updates, or other data that are later retrieved and sent to said Input Section.

31. A process for the simultaneous storage and play back of multimedia data, comprising the steps of:

providing a physical data source, wherein said physical data source accepts broadcast data from an input device, parses video and audio data from said broadcast data, and temporarily stores said video and audio data;

providing a source object, wherein said source object extracts video and audio data from said physical data source;

providing a transform object, wherein said transform object stores and retrieves data streams onto a storage device;

wherein said source object obtains a buffer from said transform object, said source object converts video data into data streams and fills said buffer with said streams;

wherein said source object is automatically flow controlled by said transform object;

providing a sink object, wherein said sink object obtains data stream buffers from said transform object and outputs said streams to a video and audio decoder;

wherein said decoder converts said streams into display signals and sends said signals to a display;

wherein said sink object is automatically flow controlled by said transform object;

providing a control object, wherein said control object receives commands from a user, said commands control the flow of the broadcast data through the system; and

wherein said control object sends flow command events to said source, transform, and sink objects.

32. An apparatus for the simultaneous storage and play back of multimedia data, comprising:

a module for accepting television (TV) broadcast signals, wherein said TV signals are based on a multitude of standards, including, but not limited to, National Television Standards Committee (NTSC) broadcast, PAL broadcast, satellite transmission, DSS, DBS, or ATSC;

a module for tuning said TV signals to a specific program;

at least one Input Section, wherein said Input Section converts said specific program to an Moving Pictures Experts Group (MPEG) formatted stream for internal transfer and manipulation;

a Media Switch, wherein said Media Switch parses said MPEG stream, said MPEG stream is separated into its video and audio components;

a module for storing said video and audio components on a storage device;

at least one Output Section, wherein said Output Section extracts said video and audio components from said storage device;

wherein said Output Section assembles said video and audio components into an MPEG stream;

wherein said Output Section sends said MPEG stream to a decoder;

wherein said decoder converts said MPEG stream into TV output signals;

wherein said decoder delivers said TV output signals to a TV receiver; and

accepting control commands from a user, wherein said control commands are sent through the system and affect the flow of said MPEG stream.

33. The apparatus of claim **32**, wherein said Input Section directs said MPEG stream to the destination indicated by said control commands.

34. The apparatus of claim **32**, wherein said Output Section extracts said video and audio components from the storage device indicated by said control commands.

35. The apparatus of claim **32**, further comprising:

a module for creating custom video output sequences, wherein said sequences are specified by a user or program control.

36. The apparatus of claim **32**, wherein the storing and extracting of said video and audio components from said storage device are performed simultaneously.

37. The apparatus of claim **32**, wherein said Media Switch calculates and logically associates a time stamp to said video and audio components.

38. The apparatus of claim **32**, wherein said Media Switch extracts time stamp values from a digital TV stream and logically associates said time stamp values to said video and audio components.

39. The apparatus of claim **32**, further comprising:

a module for placing said video component into a circular video buffer;

a module for posting an event in a circular event buffer, wherein said event contains an indication that a video component was found and the location of said video component in said circular video buffer; and

a module for sending notice of said event posting.

40. The apparatus of claim **32**, further comprising:

a module for placing said audio component into a circular audio buffer;

a module for posting an event in a circular event buffer, wherein said event contains an indication that an audio component was found and the location of said audio component in said circular audio buffer; and

a module for sending notice of said event posting.

41. The apparatus of claims **39** or **40**, further comprising:

a module for receiving said notice;

a module for retrieving said event posting from said event buffer; and

a module for indexing into the appropriate buffer indicated by the type and location information in said event buffer.

42. The apparatus of claim **41**, further comprising:

a module for generating a buffer containing the logical audio or video segments in order, including ancillary information, wherein each of said logical segments points to the appropriate circular buffer location where corresponding audio or video components have been placed.

43. The apparatus of claim **32**, further comprising:

a module for increasing the decoder system clock rate for fast playback or fast reverse playback.

44. The apparatus of claim **32**, further comprising:

a module for decreasing the decoder system clock rate for slow playback or slow reverse playback.

45. The apparatus of claim **32**, further comprising:

a module for combining system audio cues and on-screen displays with said TV output signals.

46. The apparatus of claim **32**, further comprising:

a module for decoding the Vertical Blanking Interval (VBI) data or private data channel information from said TV signal; and

a module for examining said data to determine the starting or ending indicators of a specific program.

47. The apparatus of claim **32**, further comprising:

a module for scanning the words contained within the closed caption (CC) fields to determine program starting and ending times, wherein particular words or phrases are used to trigger the recording of a specific program and wherein the CC information is preserved in time synchronization with the audio and video, and can be correctly presented to the viewer when the stream is displayed.

48. The apparatus of claim **47**, further comprising:

a module for performing a specific action when a specific word is found in said CC information.

49. The apparatus of claim **32**, wherein said Media Switch has a data bus connecting it to a CPU and DRAM.

50. The apparatus of claim **32**, wherein said Media Switch shares an address bus with a CPU and DRAM.

51. The apparatus of claim **32**, wherein said Media Switch operates asychronously and autonomously with a CPU.

52. The apparatus of claim 32, wherein said storage device is connected to said Media Switch.

53. The apparatus of claim 32, wherein said Media Switch allows the CPU to queue up Direct Memory Access (DMA) transfers.

54. The apparatus of claim 32, further comprising:

a multimedia recording device, including, but not limited to, a Video Cassette Recorder (VCR) or a Digital Video Disk-Random Access Memory (DVD-RAM) device, wherein said recording device is attached to the output side of said decoder, allowing said user to record said TV output signals.

55. The apparatus of claim 54, wherein said user queues up programs from said storage device to be stored on said recording device.

56. The apparatus of claim 54, wherein said user sets time schedules for said programs to be sent to said recording device.

57. The apparatus of claim 54, wherein title pages may be sent to said recording device before sending a program to be stored on said recording device.

58. The apparatus of claim 54, wherein a program that is longer in duration than a magnetic tape in said recording device allows, is sped up to fit within the desired time limit.

59. The apparatus of claim 54, wherein a program that is longer in duration than a magnetic tape in said recording device allows, has frames dropped from it to fit within the desired time limit.

60. The apparatus of claim 54, wherein the output of said recording device is routed to said Input Section, allowing said recording device to act as a storage back up system, said recording device accepts overflow storage, TV programs, software updates, or other data that are later retrieved and sent to said Input Section.

61. An apparatus for the simultaneous storage and play back of multimedia data, comprising:

a physical data source, wherein said physical data source accepts broadcast data from an input device, parses video and audio data from said broadcast data, and temporarily stores said video and audio data;

a source object, wherein said source object extracts video and audio data from said physical data source;

a transform object, wherein said transform object stores and retrieves data streams onto a storage device;

wherein said source object obtains a buffer from said transform object, said source object converts video data into data streams and fills said buffer with said streams;

wherein said source object is automatically flow controlled by said transform object;

a sink object, wherein said sink object obtains data stream buffers from said transform object and outputs said streams to a video and audio decoder;

wherein said decoder converts said streams into display signals and sends said signals to a display;

wherein said sink object is automatically flow controlled by said transform object;

a control object, wherein said control object receives commands from a user, said commands control the flow of the broadcast data through the system; and

wherein said control object sends flow command events to said source, transform, and sink objects.

* * * * *